# Лабораторная работа 5

### Задача:

1. Реализовать веб-сервер на Golang, который будет отдавать контент и страницы вашего блога по следующим правилам:

1. Главная страница блога отдаётся по пути /home - Главная страница это index.html
2. Страница поста должна отдаваться через /post - страница поста сейчас это post.html со статьей The Road Ahead
3. Статический контент блога должен отдаваться по пути /static - статический контент, это css файлы и картинки

2. Данные о постах Featured Posts и Most Recent теперь должны быть захардкожены в Golang, а главная страница должна шаблонизироваться через шаблоны Go.

Пример захардкоженных данных для статей в блоке Featured Posts

type featuredPostData struct {

Title string

Subtitle string

ImgModifier string

Author string

AuthorImg string

PublishDate string

}

func **featuredPosts**() []featuredPostData {

return []featuredPostData{

{

Title: "The Road Ahead",

Subtitle: "The road ahead might be paved - it might not be.",

ImgModifier: "featured-post\_\_background\_the-road-ahead",

Author: "Mat Vogels",

AuthorImg: "static/img/mat-vogels.png",

PublishDate: "September 25, 2015",

},

{

...

},

}

}

В результате в браузере у вас страница будет выглядеть точно так же(те же данные и оформление страницы), но уже список постов и их наполнение храниться и шаблонизироваться через Golang.

Пример шаблона в html, в котором поддерживается отображение списка постов

<div>

<h2>Featured Posts</h2>

{{ range .Posts }}

<div>

<h3>{{ .Title }}</h3>

<h4>{{ .Subtitle }}</h4>

<span>{{ .Author }}</span>

<span>{{ .PublishDate }}</span>

</div>

{{ end }}

</div>

### Структура проекта

С этой лабораторной вступает в силу требование по структуре проекта

<https://docs.google.com/document/d/19TCEDb4ss-6LGfsZFdXlPX6tcinB3lmmPgpJwr4P7e8/edit?usp=sharing>

Так же в структуре указаны требования к названию проекта.

### Соглашения по кодированию в Go

Документ с соглашениями - https://docs.google.com/document/d/1Vf7gtb4KuFUTHtaXrRR5EmNYjDcpjVkk9IXm2J5o4qE/edit?usp=share\_link

### Начало в Go

Любой Go-проект начинается с инициализации go.mod (файл с метаинформацией проекта).

go mod init blog

Дальше можно создавать файлы и скомпилировать проект

### Компиляция и запуск Go программы

Перед первым запуском нужно отключить CGO для компилятора, нужно выполнить в консоли:

go env -w CGO\_ENABLED="0"

Скомпилировать программу и запустить можно через одну команду, запускать нужно всегда из корня проекта:

go run <путь до файлов в папке cmd>

go run cmd/blog/main.go cmd/blog/handlers.go

### 

### Пример

По ссылке лежит небольшой проект на Go, в котором использованы основные инструменты для решения задачи

https://drive.google.com/file/d/1Wx5To2CnlXp-e4WX-0xaEmwJ1\_iHVLWO/view?usp=share\_link